AD-A120 191 AERONAUTICAL RESEARCH LABS MELBOURNE (AUSTRALIA) F/G 9/2
COMPUTER GRAPHIC PRESENTATION OF CONVENTIONAL COCKPIT INSTRUMEN--ETC(U)
NOV 81 H A THELANDER, R J ROSSA
UNCLASSIFIED ARL-SYS-NOTE-82 NL

1 OF 1
AD A
120191

END
DATE
FILMED
11-82
DTIC

(12)

ARL-SYS-NOTE-82

AR-002-320

AD A120191

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORIES

MELBOURNE, VICTORIA

SYSTEMS NOTE 82

COMPUTER GRAPHIC PRESENTATION OF CONVENTIONAL COCKPIT INSTRUMENTS

by

H. A. THELANDER and R. J. ROSSA

Approved for Public Release

DTIC FILE COPY

A

COPY No 19

NOVEMBER, 1981

82 10 14 002

AR-002-320

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORIES

SYSTEMS NOTE 82

COMPUTER GRAPHIC PRESENTATION OF CONVENTIONAL COCKPIT INSTRUMENTS

by

H. A. THELANDER and R. J. ROSSA

SUMMARY

Conventional aircraft cockpit instruments, presented on a computer generated colour display, are discussed in the context of research flight simulation, and their general characteristics in this application are established. The hardware and software involved in a realization of a simulator cockpit instrument panel using this approach is described, and the significant problem areas and solutions achieved are highlighted. Detailed discussion of two of the more interesting instruments is provided to illustrate the methods. Analysis of subjective assessments and timing measurements confirms the success of the work.

POSTAL ADDRESS: Chief Superintendent, Aeronautical Research Laboratories,
Box 4331, P.O., Melbourne, Victoria, 3001, Australia.

DOCUMENT CONTROL DATA SHEET

Security classification of this page: Unclassified

1. Document Numbers (a) AR Number: AR-002-320 (b) Document Series and Number: Systems Note 82 (c) Report Number: ARL-Sys-Note-82	2. Security Classification (a) Complete document: Unclassified (b) Title in isolation: Unclassified (c) Summary in isolation: Unclassified
3. Title: Computer Graphic Presentation of Conventional Cockpit Instruments	
4. Personal Author(s): H. A. Thelander R. J. Rossa	5. Document Date: Written July, 1981 Published November, 1981
	6. Type of Report and Period Covered:
7. Corporate Author(s): Aeronautical Research Laboratories	8. Reference Numbers (a) Task: DST 80/098 (b) Sponsoring Agency: DSTO
9. Cost Code: 716170	
10. Imprint: Aeronautical Research Laboratories	11. Computer Program(s) (Title(s) and language(s)): RGRPAC—MACRO-10 RGRP11—MACRO-11

12. Release Limitations (of the document):
Approved for public release

12-0. Overseas:	N.O.		P.R.	1	A		B		C		D		E	

13. Announcement Limitations (of the information on this page):
No Limitations

14. Descriptors:		15. Cosati Codes:
Flight simulators	Computer graphics	0104
Display devices	Aircraft instruments	0902
Cockpits		

16. *ABSTRACT*

Conventional aircraft cockpit instruments, presented on a computer generated colour display, are discussed in the context of research flight simulation, and their general characteristics in this application are established. The hardware and software involved in a realization of a simulator cockpit instrument panel using this approach is described, and the significant problem areas and solutions achieved are highlighted. Detailed discussion of two of the more interesting instruments is provided to illustrate the methods. Analysis of subjective assessments and timing measurements confirms the success of the work.

CONTENTS

Page No.

1. INTRODUCTION 1

2. APPLICATION REQUIREMENTS 1

2.1 General Display Arrangement 2

2.2 Instrument Characteristics 2

2.2.1 Counter—Pointer Instruments 2

2.2.2 Altitude Indicator 2

2.2.3 Horizontal Situation Indicator 4

2.2.4 Other Flight Instruments 4

2.2.5 I.L.S. Indicator 6

2.2.6 Annunciators 6

3. APPROACH 6

3.1 Raster Graphics Hardware Subsystem 7

3.1.1 Monitors 7

3.1.2 Controller and Memory 7

3.1.3 Palette 8

3.1.4 Data Formats 8

3.2 Software Systems 9

3.3 RGRPAC 9

3.3.1 Initialization and Termination 10

3.3.2 Graphics Primitives 10

3.3.3 Text Primitives 11

3.3.4 Conversion of Coordinates from Millimetres to Pixels 11

3.3.5 Display Code Storage 11

3.4 DL10 Communications 12

3.4.1 Loading the PDP-11/34A Program 12

3.4.2 Transmission of Image Data 12

3.4.3 Program Control 13

3.4.4 System Hazard 13

3.5 RGRP11 13

3.5.1 Motion 14

3.5.2 Area Fill 14

3.5.3 Trigonometric Functions 15

3.5.4 Text 16

4. IMPLEMENTATIONS 16

4.1 Altitude Indicator 16

4.1.1 Horizon and Sky 16

4.1.2 Pitch Scale 18

4.1.3 Bank Pointer 18

4.1.4 Aircraft Datum 18

4.1.5 Order of Operations 19

4.2 Horizontal Situation Indicator 19

4.2.1 Compass Card 19

4.2.2 Other Features 19

5. RESULTS 22

6. DISCUSSION 24

REFERENCES

APPENDIX

DISTRIBUTION

1. INTRODUCTION

The requirements for the presentation of aircrew cockpit displays in a research manned flight simulator are biased strongly towards flexibility and versatility in information content and format. This is in contrast to the requirements in specific-to-type training simulators, where fidelity of representation of the aircraft systems and characteristics is generally considered to be vital to correct transfer of training. Computer-generated colour raster displays have been examined for the research application, and their potential benefits established [1].

This note describes the presentation of a conventional cockpit instrument panel on a colour raster display. The application is in a Multi-Crew-Aircraft Simulator (MCAS), as the first simulator implementation in a planned programme of manned flight simulation research.

A conventional instrument panel was chosen for this first stage in preference to the more innovative approaches to cockpit instrumentation that will be implemented in the future. A conventional instrument panel has the immediate advantage of being recognizable, and therefore usable, by current aircrew; avoiding the need for extensive experimental subject training. The general characteristics of the conventional panel layout and individual instruments are well established, by standard in many cases, and this means that less design effort is required, leaving a maximum of resources for the implementation. The conventional panel is mostly non-integrated, in that a separate instrument is provided for the majority of the parameters presented. In display processing terms this makes it a more severe test of the system than the integrated innovative displays, and its successful realization is thus a good test of the potential of the approach.

Manned Flight Simulation research at A.R.L. is in its early stages, with the MCAS being used as a technology investigation and demonstration vehicle as well as for applications-oriented work. Parallel developments to those reported in this Note cover computer generated wide angle colour visuals for the external environment, electro-pneumatic motion base, electric control force-feel *G*-seat, and noise environment generation. The aim is to provide a versatile research and study capability, adaptable to the representation of the essential features of a wide variety of aircraft and systems. Initial applications studies will include investigation of predictive displays in flight operations, aircrew workload and task sharing, and examination of simulator cue requirements.

The next chapter addresses the instrument panel display requirements in functional terms and then in the detail of the characteristics of the individual instruments. Chapter 3 covers the broad aspects of the approach taken in the work, placing it in the context of the complete MCAS development, and describing the software and hardware systems and their integration, followed by a discussion of the implementation in detail, showing how the graphics primitive functions are used to construct the display and give it motion. Chapter 4 provides a detailed examination of the algorithms and programming for two of the instruments, as a practical illustration of the process of generating new or modified forms. Chapter 5 presents the results of the work, with a photograph of the actual display, and Chapter 6 discusses these and future work in the field.

An appendix contains a detailed description of the alphanumeric text generation processes used in the application.

2. APPLICATION REQUIREMENTS

The MCAS, in which the work reported here is used, has a two-seat side-by-side flight deck of utility transport aircraft size. A centre pedestal mounts engine, flap, undercarriage and rudder trim controls; a flight control yoke and rudder pedals are installed in the left-hand (P1) position. The instrument panel space holds three 480 mm (diagonal) colour television monitors, mounted side-by-side across its full width. All cockpit instrumentation for piloting a twin turbo-prop light utility transport is duplicated on the left and right monitors. In this first imple-

mentation the centre monitor is not used, resulting in some crowding of the display. In later stages of simulator development, the engine instrumentation in particular will be relocated from the P1 monitor.

2.1 General Display Arrangement

The general arrangement of the display is illustrated in Figure 1 (drawn approximately half scale). The primary flight instruments are arranged in the standard conventional 'Basic T' pattern [2, 3], with horizontal situation display forming the leg of the T below, in line from left to right, an airspeed indicator, attitude indicator and altimeter. A vertical speed indicator below the altimeter completes the pattern. Outside the Basic T are representations of a turn and slip indicator and of an ILS crossed pointer indicator. Engine instruments for free shaft turbine engines [4] are arranged in two adjoining columns of five circular dials. These latter are placed to the left of the display to permit the Basic T to be located near the pilot's centre line. In the simulator they can also be seen from the P1 seat at the left of the P2 (right hand) display, nearer their familiar cockpit central location.

Digital clocks and an event timer, caution annunciators, and communications channel readouts fill up the top of the display, with more annunciators and strip type indicators for trims, flaps and undercarriage located at the bottom. A column of eight boxes on the right of the display is used to define the functions of eight programmable push-buttons located adjacent in the bezel.

2.2 Instrument Characteristics

The displayed instruments do not replicate specific devices, but conform to the general standards for cockpit instruments [5]. Their arrangement and general characteristics are familiar to pilots. Salient details are given below.

2.2.1 Counter-Pointer Instruments

Counter-pointer instruments are used for airspeed indicator, altimeter, and the ten engine instruments. Each consists of a circular dial inscribed with fixed scale marks and numbers, a centrally-pivoted rotating pointer, and a rectangular window in which is displayed the numeric value of the parameter measured. The format combines the precision of the counter with the analogue pointer's ready indication of change.

The pointers are drawn as triangles, rather than the elongated, pointed-end rectangles shown in the standard. As well as being computationally simpler, triangular pointers avoid parallel steps-and-stairs aberrations which are quite distracting on the rectangles. The larger instruments--airspeed indicator and altimeter--allow multiple revolutions of the pointer, and therefore have scale markings running from zero (at the top) clockwise to nine. Whilst their dimensions have been arranged so that the pointers do not obscure the scale marks and numbers, inevitably the pointers do cross the counter windows. For enhanced legibility the counters are shown 'in front' of the pointers, something not achieved in real instruments.

The engine instruments are smaller, and their pointers are restricted to 270° motion (from zero at the top to twelve on the scale at the 9-o'clock position) or less. The pointers do not obscure the scale graduations, and the counters are located out of the areas swept by the pointers.

2.2.2 Altitude Indicator

The altitude indicator presents the pilot with an 'inside-out' analogue of the aircraft's pitch and bank. Many forms are in current service, employing spherical and roller-blind mechanizations. The instrument displays a moving horizon, the sky distinguished by its light blue or grey colour from a black ground. The aircraft is symbolized in the centre of the instrument. The standard [6] requires the bank pointer to be placed *in the lower part of the indicator*. Figure 2 shows the presentation for an aircraft pitched up 10° and banked 15° to the right (i.e. right wing down).

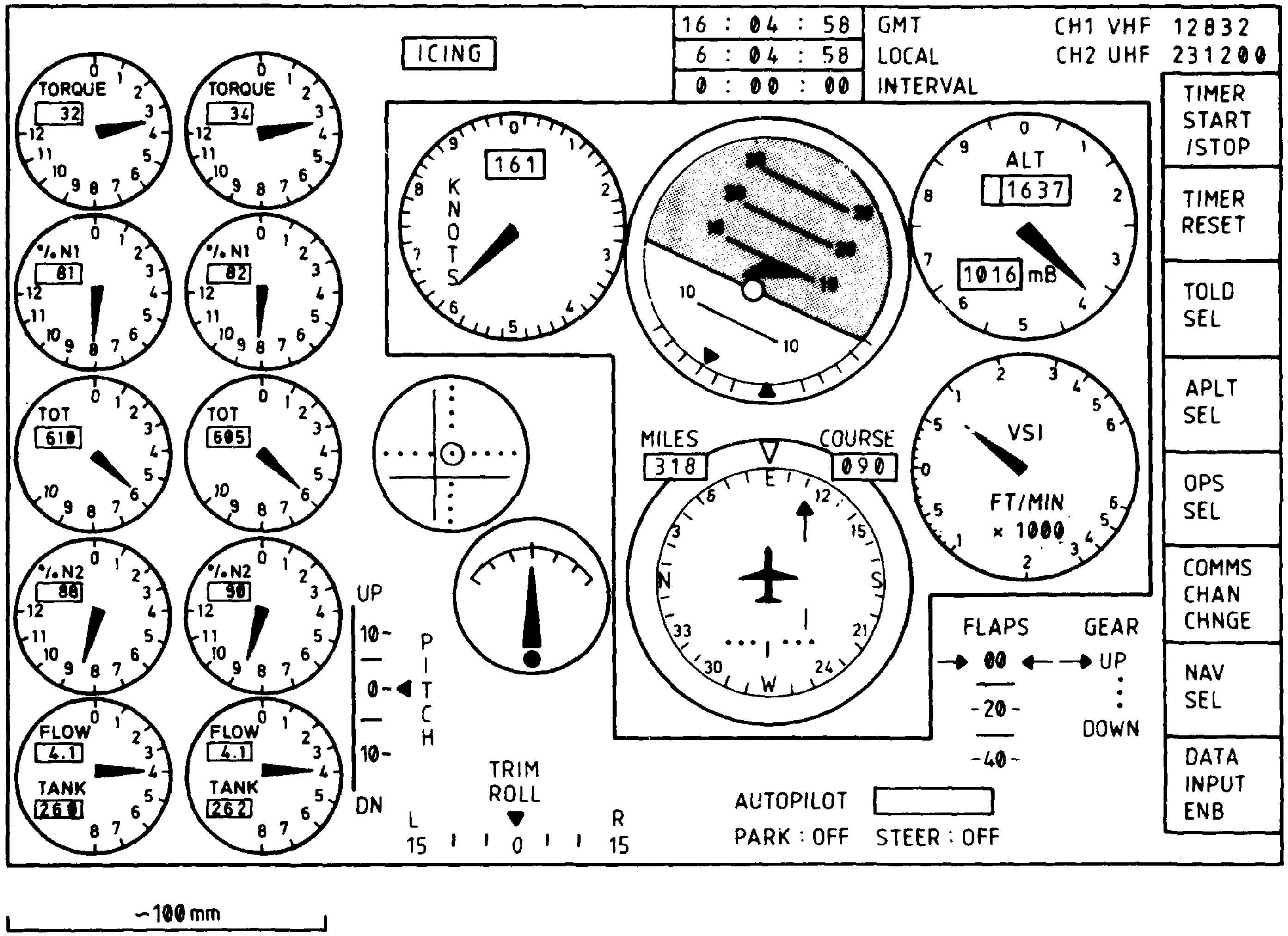


FIG. 1: DISPLAY ARRANGEMENT

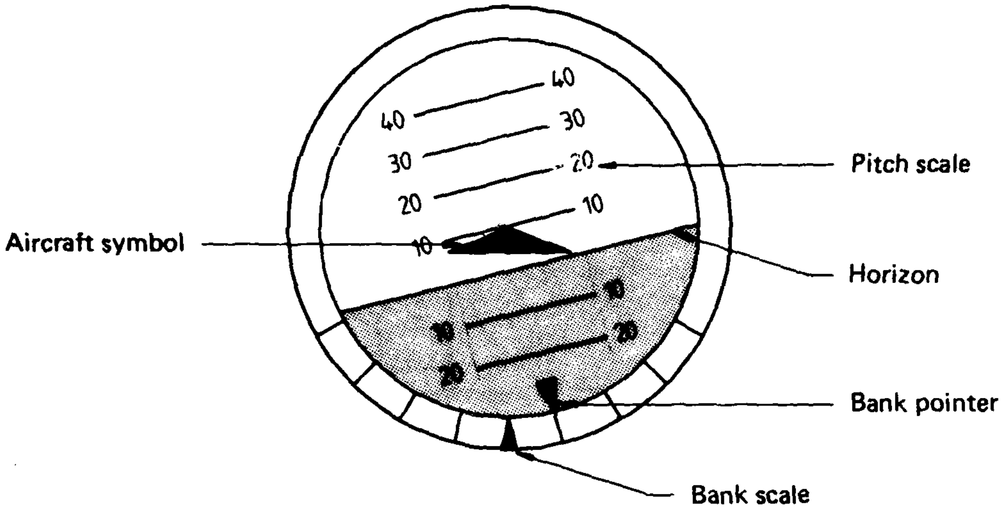


FIG. 2: ATTITUDE INDICATOR SHOWING 10° PITCH UP, 15° BANK TO RIGHT

The 'gearing' of the pitch scale may be quite coarse, as for example in instruments utilizing a spherical 'bijou-ball'. Generally the gearing is finer, showing 60°, or less, pitch range from top to bottom, where a 'roller-blind' mechanism is used. In these cases it is usual to limit the horizon excursion so that it is displayed even at extreme pitch angles, allowing the pitch scale to slip when this occurs.

2.2.3 Horizontal Situation Indicator

The horizontal situation indicator displays a 'heading up' lubber line and moving compass card; a miniature aircraft is fixed at its centre [7]. To this layout are added various other features depending on the application: a To-From arrow indicating aircraft position relative to a selected radio facility being most common. Digital readouts give range and bearing information. Figure 3 shows the presentation for an aircraft heading 20° east of north, converging from the left on to a selected course of 330°, at a range of 28 (n.mi.).

2.2.4 Other Flight Instruments

The other flight instruments required in the MCAS are a vertical speed indicator (VSI) and a turn-and-slip indicator (TSI). The former is part of the Basic-T, and the latter useful for instrument flight and procedural manoeuvres.

The VSI [8] employs an analogue pointer in a circular scale, with the zero at the 'nine-o'clock' position. Climb is indicated by a clockwise (initially upwards) pointer motion. A symmetrical non-linear scale is employed, calibrated in thousands of feet per minute to six thousand feet per minute vertical speed limits.

The form of the TSI is indicated in Figure 4.

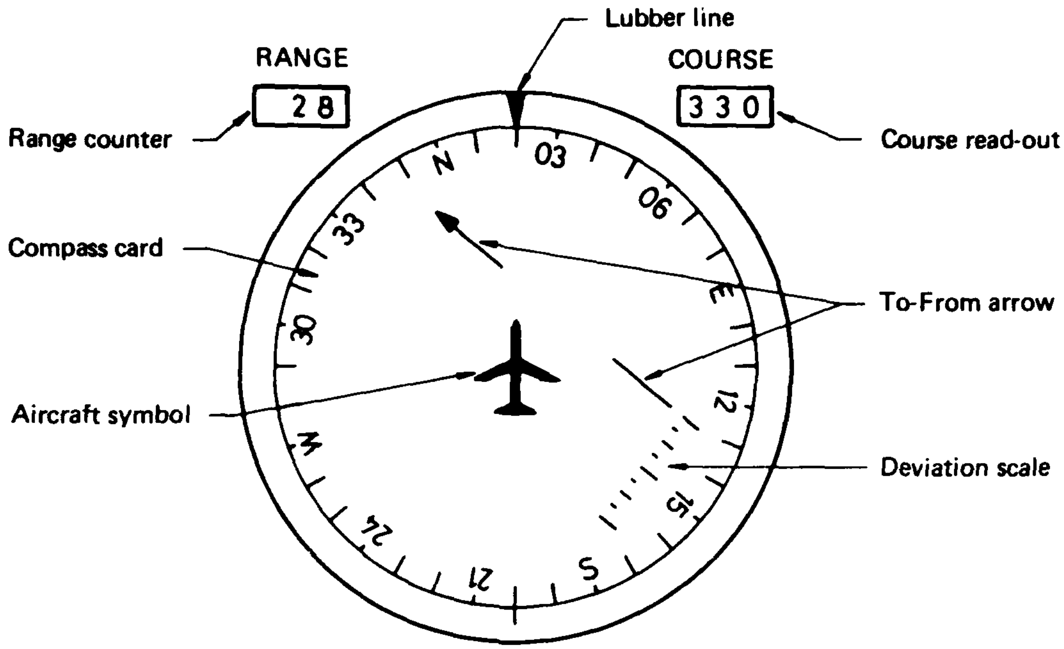


FIG. 3: HORIZONTAL SITUATION INDICATOR (see text)

A turn pointer, pivoted below the centre of the circular dial, moves over a turn rate scale. By convention a rate 1 turn would achieve 360° in two minutes, and rate 2, 360° in one minute. Below the pointer is a yaw 'bubble', actually a bubble in a fluid-filled tube in early instruments, measuring side force and used to maintain turn coordination.

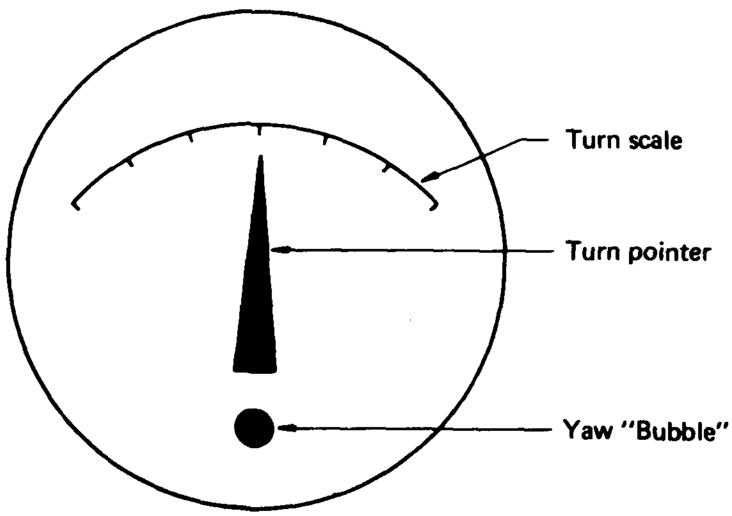


FIG. 4: TURN AND SLIP INDICATOR

2.2.5 ILS Indicator

Instrument landing system (ILS) glideslope deviation is given by a crossed-pointer indicator, showing lateral and vertical deviation. The horizontal pointer is below the instrument centre when the aircraft is above the glideslope, and the vertical pointer to the right when the aircraft deviates to the left. The pilot's correct action is therefore to 'fly' the instrument centre towards the pointers.

2.2.6 Annunciators

Figure 1 shows the remainder of the arrangement of the panel. Triangular pointers move past linear scales to indicate pitch and roll trim settings (rudder trim is located on the cockpit centre pedestal), and flap and (landing) gear position. Caution warnings, communications channels, nosewheel steering, parking brake setting and autopilot status are displayed in text. A digital display of Greenwich Mean, Local, and elapsed times is provided.

3. APPROACH

An overview of the MCAS systems is given in Figure 5.

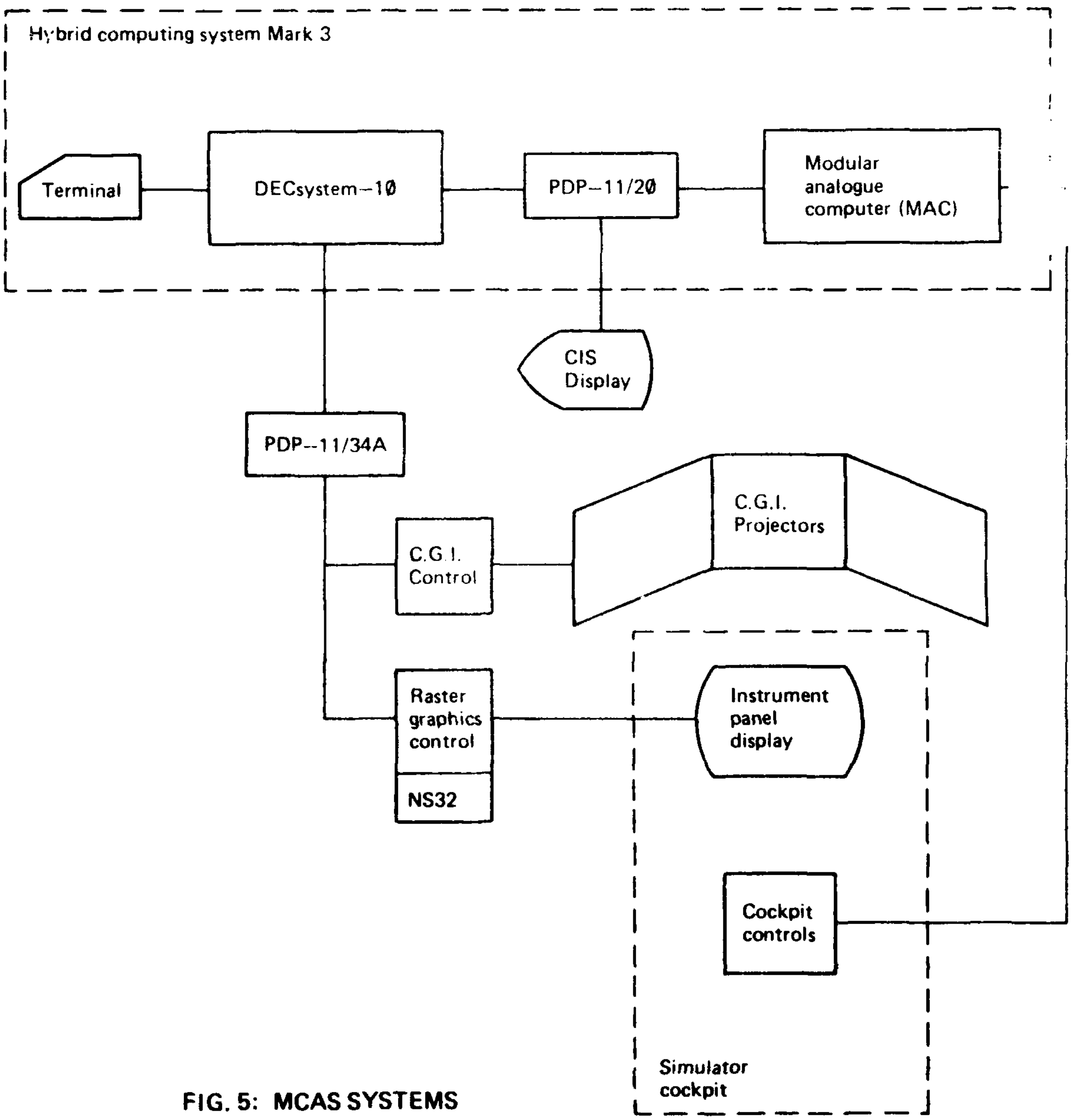


FIG. 5: MCAS SYSTEMS

Real-time dynamic simulation of the aircraft's flight is performed by the A.R.L. Hybrid Computing System Mark 3 (HCS3). The problem is conventionally partitioned, with the faster dynamics being computed by the analogue part, the Modular Analogue Computer (MAC); and the slower by the digital, the DECsystem-10. The DECsystem-10 terminal, used for system control, is located with the CIS refreshed graphics display, and repeater monitors of the instrument panel and CGI (see below) centre screen, at the simulator operator station.

A CGI (Computer General Imagery) subsystem presents a simulated external visual environment on three back-projection screens located around the front of the simulator cockpit. In the cockpit are the instrument panel display and the pilots flight controls. The flight controls are equipped with analogue position and force sensors, from which signals are fed to the MAC. The position signals, in particular, are then input to the aircraft dynamic model. The MAC also computes force-feel signals which drive electric servo-motors linked to the controls.

Separate from the HCS3 hardware, the PDP-11/34A minicomputer is interfaced via the DL10 data link to the DECsystem-10. The DL10 provides a window in UNIBUS address space through which the PDP-11/34A can refer to programs or data actually located in the core storage of the DECsystem-10, and control functions giving the DEC-system-10 control of the PDP-11/34A processor. The PDP-11/34A serves as a communications link, allowing transfer of CGI data computed by the DECsystem-10 via a multiple direct memory access process to the CGI controller where it is converted to video. This places only a small interrupt processing load on the PDP-11/34A, and a small direct memory access load on its UNIBUS.

The main processing function of the PDP-11/34A in the MCAS is the maintenance of the instrument panel display. Instrument panel data are placed in assigned locations in the DL10 window, whence they are read by the program in the PDP-11/34A and used in updating the display.

3.1 Raster Graphics Hardware Subsystem

3.1.1 Monitors

The monitors used for display of the instrument panels are 'National' (Matsushita Electric Trading Co. Ltd.) Model TC 2002 48 cm (diagonal) colour television receivers modified to accept separate red, green and blue (RGB) input video signals and composite sync. The picture displayed is composed of 512 × 512 pixels, filling a rectangle approximately 400 mm wide by 273 mm high on these monitors. The pixels themselves are thus approximately 0·8 mm wide by 0·5 mm high. The format is described in more detail, and the consequences of the non-squareness of the pixels evaluated, in Reference [1].

3.1.2 Controller and Memory

Figure 6 shows schematically the arrangement of the components of the raster graphics subsystem. Two bits of data per pixel are stored in the National Semiconductor Corporation

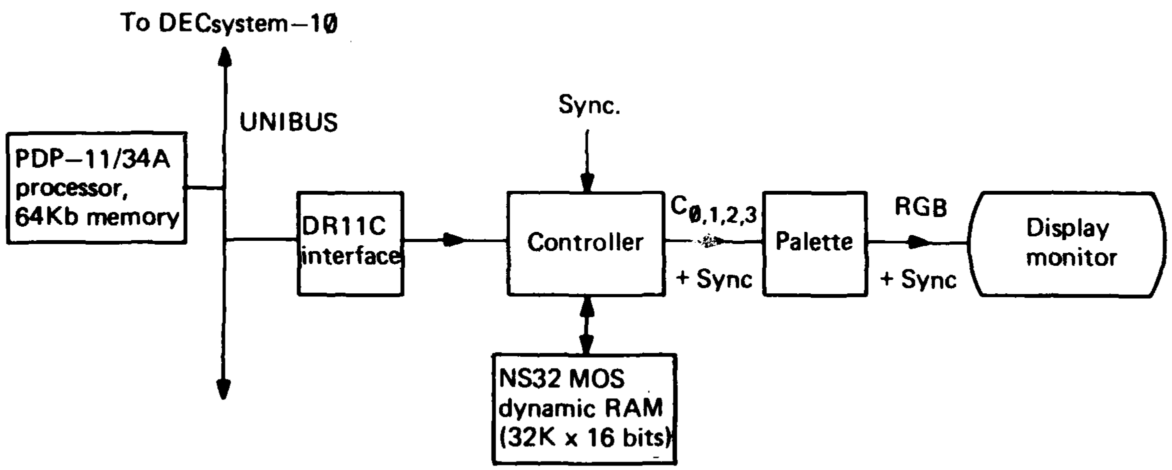


FIG. 6: RASTER GRAPHICS HARDWARE SUBSYSTEM

Model NS32 dynamic MOS RAM. Data are loaded into it by the controller in response to commands transmitted to it by the processor through the DR11C register interface. The controller reads these out, and performs the dynamic memory refresh function, using external video synchronization signals.

3.1.3 Palette

The output from the controller is four 'colour' signals, termed Colour 0, Colour 1 Colour 2 and Colour 3. These are passed to a 'palette', providing for each Colour three slider potentiometers, one each for Red, Green and Blue. The raster graphics user can then mix any colour to correspond to each of the controller output codes. In the case of the instrument panel display reported here, the assignments are:

Colour 0	Black
Colour 1	White
Colour 2	Red
Colour 3	- Green.

The output from the palette is RGB video signals plus composite sync.

3.1.4 Data Formats

The raster graphics controller accepts and processes the commands sent to it by the PDP-11/34A processor. Figure 7 shows the command formats. The controller contains internal x- and y-coordinate pixel address registers and a colour register, respectively nine, nine, and four bits wide. The coordinates range from 0 to 511, with pixel (0, 0) at screen bottom left. The colour register contains two two-bit-fields, one of which is used to load data for 'odd' pixels, the other for 'even' pixels. Even pixels are those whose coordinates, least significant bits are equal. The odd and even pixels constitute (mutually exclusive) chequerboards.

The controller accepts commands for setting either the x- or the y-coordinate register, with the option of thereupon loading the appropriate (odd or even) colour code into the pixel then addressed. A third command allows either or both of the x- and y-coordinate registers to be incremented (by one), or untouched, or decremented (by one) with the option then of loading the appropriate colour code into the pixel at the updated address. The remaining command provides for setting the colour register, and optionally for loading all pixels with the updated even pixel colour: this last function is used to clear the refresh memory at the start of operation.

Operation	Bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Incremental Move	0	0	W					Unused					X_C	X_N	Y_C	Y_N
Set X-Coordinate	0	1	W		Unused		X_8	X_7	·	·	·	·	·	·	X_1	X_0
Set Y-Coordinate	1	0	W		Unused		Y_8	Y_7	·	·	·	·	·	·	Y_1	Y_0
Set Pixel Colour Code	1	1	S					Unused					B_O	A_O	B_E	A_E

BIT	VALUE	ACTION
X_C	1	COUNT X-COORDINATE
	0	NONE
Y_C	1	COUNT Y-COORDINATE
	0	NONE
X_N, Y_N	1	COUNT DOWN (IF COUNT BIT = 1)
	0	COUNT UP (IF COUNT BIT = 1)
$X_8 \cdots X_0$	n	NEW X-COORDINATE VALUE ($511 \geq n \geq 0$)
$Y_8 \cdots Y_0$	n	NEW Y-COORDINATE VALUE ($511 \geq n \geq 0$)
$B_E A_E$	c	COLOUR VALUE, EVEN PIXELS ($3 \geq c \geq 0$)
$B_O A_O$	c	COLOUR VALUE, ODD PIXELS ($3 \geq c \geq 0$)
W	1	WRITE PIXEL AT UPDATED COORDINATES
	0	NONE
S	1	WRITE ALL PIXELS TO UPDATED COLOUR VALUE $B_E A_E$
	0	NONE

Figure 7: Raster Graphics Controller Command Formats.

3.2 Software Systems

Software packages for the raster graphics application were developed for both the DECsystem-10 and the PDP-11/34A. As noted above, during simulator operation the latter processor is dedicated to maintaining the instrument panel display, while the DECsystem-10 is required to pass it the appropriate data from the dynamic simulation computations. This necessitated the integration of some extra functions into the HCS3 DEC-system-10 system software package LOKXIX [9, 12]. These changes did not alter the main hybrid computation package H3PAC.

To avoid burdening the PDP-11/34A operational program with the once-only executed code required for initially setting up the fixed framework of the instrument panel display, and to allow the programming of this part of the operation to be performed in a high level language (none of which are available in the PDP-11/34A used here), a DECsystem-10 software package of FORTRAN-callable graphics primitive routines was written. This package, RGRPAC, provides as well a general-purpose static raster graphics capability for use with this hardware on the DECsystem-10, and has been exploited for a number of other image reconstruction applications. RGRPAC optionally allows for storing the image data it generates as a compressed binary file on the DECsystem-10's disk storage. These data can be subsequently reloaded with no computational overhead into the raster graphics memory.

In the PDP-11/34A, a single program, RGRP11, is used both with RGRPAC to construct and reload the fixed framework of the instrument panel and with the DECsystem-10 raster graphics hybrid run time package RGRUNP in simulator operation.

The remainder of this chapter addresses the important features of the DECsystem-10 software, the inter-computer communication, and the PDP11/34A program RGRP11.

3.3 RGRPAC

RGRPAC, the DECsystem-10 static raster graphics package, is written in the MACRO-10 assembly language, and provides DECsystem-10 FORTRAN-callable subroutines for initializing

and terminating raster graphics activity, for the graphics primitive functions of point and vector drawing, and for displaying text.

3.3.1 Initialization and Termination

The initialization subroutine uses code added to the HCS3 load checking and X1X interface handler program module LOKX1X [9] to obtain a free 512-word memory 'page' for use as the PDP-11/34A UNIBUS window by the DL10. Arguments of the initialization call specify whether, and if so by what name, a file containing the display code generated during the program's execution is to be stored, compressed, on the DECsystem-10's disks. Buffers necessary for this are constructed. The initialization routine contains a guard mechanism designed to ensure that only one user program at a time can gain access to the DL10 and thereby the PDP-11/34A.

When raster graphics processing is done, the termination routine completes the transmission of any data remaining to be sent either to the DL10 or to the saved code file. The DL10 is cleared and the window page freed for other use.

3.3.2 Graphics Primitives

RGRPAC contains subprograms for 'drawing', that is entering into the raster graphics memory, single points (pixels) addressed by their *x*- and *y*-coordinates, and vectors of points from the most recently drawn point to a specified end point. The vector function could be performed by the user program, but in the interests of efficiency it was coded into RGRPAC as a primitive. The colour code of the point or vector drawn is specified as an argument of the subprograms, as an integer in the range zero to fifteen (decimal), directly translating the pixel colour code shown in Figure 7. Codes 0, 5, 10 and 15 select the 'pure' colours (odd and even pixels of the same colour), the other codes select 'composite' colours. With some care in adjusting the palette for the pure colours it is possible to produce ten useful, distinguishable colours for filling areas, albeit at a lower resolution.

The vector algorithm adopted is similar to the 'Simple DDA (Digital Differential Analyser)' described by Newman and Sproull [10], using a modulus arithmetic division implemented in additions, subtractions and comparisons. It produces a best-fit approximation to the straight line joining its endpoints using a minimum number of steps and in a form compatible with the incremental move command format of the controller.

A PASCAL implementation of the vector algorithm is as follows:

```
procedure VECTOR (x0, y0, x1, y1, colour: integer);
    const incx = 10B; decx = 14B; incy = 2B; decy = 3B; writepixel = 20000B;
var count, acc, major, minor, stepmajor, stepboth: integer;
begin
    Setcolour (colour);
    Draw (x0, y0);
    if abs (x1 - x0) >= abs (y1 - y0)
        then begin major := abs (x1 - x0); minor := abs (y1 - y0);
            if x1 >= x0
                then stepmajor := incx + writepixel
                else stepmajor := decx + writepixel;
            if y1 > y0
                then stepboth := incy + stepmajor
                else stepboth := decy + stepmajor
            end
```

```
        else begin major: = abs (y1 - y0); minor: = abs (x1 - x0);
            if y1 > y0
                then stepmajor: = incy + writepixel
                else stepmajor: = decy + writepixel;
            if x1 > x0
                then stepboth: = incx + stepmajor
                else stepboth: = decx + stepmajor
        end;
    acc: = major div 2;
    for count: = 1 to major
        do begin acc: = acc - minor;
            if acc > 0
                then Emit (stepmajor)
                else begin Emit (stepboth);
                        acc: = acc + major
                    end
            end
end;
```

3.3.3 Text Primitives

RGRPAC includes subprograms for drawing strings of text and single numeric digits, at a position and in a colour specified in the call. Two complete (95-character ASCII) text fonts and two numeric fonts were generated using the DECsystem-10 and the DEC type 338 interactive display. The fonts are stored in RGRPAC as strings of display code. Appendix A gives details of the fonts.

3.3.4 Conversion of Coordinates from Millimetres to Pixels

The use of non-square pixels in the raster graphics system makes the specification of screen coordinates in pixels inconvenient. To assist in laying out the display format, and as an aid to such processes as drawing circles, which is done by computing

$$x = x_0 + r\cos\theta$$

$$y = y_0 + r\sin\theta$$

for values of θ in the range 0 to 2π, RGRPAC contains INTEGER FUNCTION subprograms for converting REAL x- and y-coordinate arguments in millimetres to pixels. The conversion multiplication factors are 1·25 for x and 1·875 for y: these numbers have the exact binary representations $1{\cdot}01_2$ and $1{\cdot}111_2$, so that implementing equivalent conversions in the PDP-11/34A is straightforward (see below).

3.3.5 Display Code Storage

An argument passed to the RGRPAC initialization routine allows the user to specify that the display code generated in subsequent calls to RGRPAC routines be stored in a file on the DECsystem-10's disks. If this option is exercised, then the file name to be used is also specified at that time. The sixteen-bit display command words generated by RGRPAC and transmitted

through the DL10 to the PDP-11/34A are then also written, right-justified, in successive (36-bit) words of the binary mode file. A simple compression is achieved by suppressing successive identical commands, using the left-half of the 36-bit DECsystem-10 word to carry a repetition count. The degree of compression achieved is highly dependent on the nature of the picture: for the main application, the instrument panel background, the compression well exceeds the two-to-one that would result from byte packing, that is if left- as well as right-halves of the DECsystem-10 words were used for storing commands.

3.4 DL10 Communications

The DL10 interface is part of the ARL DECsystem-10. Up to four PDP-11 minicomputers may be connected to its UNIBUS Ports, where they can be controlled by the DECsystem-10, and given read, write and execute access to code and data in a variety of formats in the DECsystem-10's core. The DL10 provides a bi-directional priority interrupt capability, allowing the DECsystem-10 to interrupt any PDP-11 that has enabled the interrupts, and any PDP-11, whose Port has been enabled, to interrupt the DECsystem-10. Unique among all PDP-11 peripherals, the DL10 interrupts in the PDP-11s have a priority assignable by program.

The DECsystem-10 uses the DL10 and the PDP-11/40 attached to its first Port for control of its network of interactive user terminals. This is managed by the DECsystem-10 operating system (TOPS-10), which is responsible for processing DL10 interrupts. Consequently, the DECsystem-10 interrupt request feature can not be used by the raster graphics PDP-11/34A with TOPS-10 unmodified. Rather than implementing appropriate, probably extensive, modifications to TOPS-10 it was decided to accept this restriction on the use of the DL10.

The DECsystem-10 and PDP-11 34A operate in a master-slave relationship. Communications are initiated by the DEC-system-10 loading a command and arguments in pre-assigned locations in the DL10 window and then emitting a PDP-11/34A interrupt request to the DL10. On receipt of the interrupt the PDP-11/34A dispatches to the appropriate command handler to perform the requested function, signifying completion by clearing the DL10 interrupt request flag. The DECsystem-10 meanwhile waits in a loop, testing for this flag to be cleared, and counting down a timer so as to detect failure of the PDP-11/34A.

3.4.1 Loading the PDP-11/34A Program

All software for the PDP-11/34A is maintained on the DECsystem-10 using its standard utility software and the MACY11 cross-assembler. The PDP-11/34A operates without either mass storage or a user terminal and with only volatile (MOS) memory. Programs are loaded through the DL10 into the PDP-11/34A, and their execution started, by a DECsystem-10 cross-loader, BOOT34, which is an ARL modification of the DEC utility BOOT11 for use with this hardware. In particular BOOT34 will only operate on the raster graphics subsystem PDP-11/34A and therefore cannot inadvertently affect the terminal communications system PDP-11/40.

3.4.2 Transmission of Image Data

Image data in the form of raster graphics controller command words are generated in the DECsystem-10 by RGRPAC. They are assembled in a buffer and transmitted as described above whenever the buffer fills and upon termination of RGRPAC use. A command which instructs the PDP-11/34A program to copy a specified number of words from a sequence of locations (in the DL10 UNIBUS window) starting at a specified location to a fixed, specified location in the PDP-11/34A's address space (the raster graphics controller data buffer) is used. The buffer's length is 128 words, so that the overhead in the transmission is less than 1% of that which would derive if the data were sent one word at a time.

The compressed image data stored on the DECsystem-10's disks by RGRPAC are transmitted in the same way, using the DECsystem-10 FORTRAN program PUTBAK to call a MACRO-10 assembler language subprogram package RGREST for this purpose. The PDP-11/34A software detects no difference between the two processes.

3.4.3 Program Control

The repertoire of commands passed from the DECsystem-10 to the PDP-11/34A includes a program jump instruction. In the present application this is used by the DECsystem-10 simulation program raster graphics run-time package, RGRUNP, to direct the PDP-11/34A program to commence updating the instrument panel display at the commencement of simulated flight, and to cease at the end.

3.4.4 System Hazard

During simulator operation the PDP-11/34A program enters a loop in which it successively reads data from pre-assigned locations in the DL10 window from the DECsystem-10's core memory, and uses them to update the instrument panel display. The DECsystem-10 program, using the RGRUNP package, meanwhile updates these locations with information reflecting the current state of the simulated flight. In the event of an abnormal termination of the DECsystem-10 program, the DECsystem-10 operating system may allocate the physical page of memory currently addressed by the DL10 window to the virtual address space of some other timesharing job. Naturally, the information stored in the pre-assigned locations by such an other job is highly unlikely to be sensible to the PDP-11/34A, and the resulting display will be meaningless.

However, the DL10 characteristics can cause more problems than this. It provides for 'indirect' data access by a PDP-11, whereby the contents of a location in the window accessed by the PDP-11 are interpreted as a pointer to a byte of data somewhere else, and this pointer may be incremented by the hardware before performing the requested accessing. Whether a PDP-11 access in the window is direct or indirect is determined by the information in the accessed location itself. In the circumstances of the abnormal termination, the result can be that the contents of another job's memory is corrupted by the pointer-incrementing process. It is clearly desirable to avoid abnormal program termination: note also that the conventional DECsystem-10 'control-C' method of stopping a program does not constitute an abnormal termination in this context.

3.5 RGRP11

The RGRP11 program in the PDP-11/34A manages the DEC-system-10 interrupts for data transfer and program control, maintains the instrument panel display during simulated flight, operates the raster graphics display controller and performs device control and interrupt processing for the CGI subsystem. RGRP11 is written in MACRO-11 assembler, maintained on the DECsystem-10, and assembled using the MACY11 cross-assembler. Its source is some 6000 lines long, and it occupies slightly more than 21K 16-bit words of program and data memory.

The program consists of a single main program loop, in which are calls to separate subprograms for the updating of each individual instrument in the display, preceded by code for hardware initialization. Execution of the main loop commences when the DECsystem-10 simulation program emits a program jump command at the start of simulated flight, and terminates with a jump back to the initialization procedures at the end of the flight. There are handlers for the interrupts generated by the DECsystem-10 and the CGI controller.

Each displayed instrument's updating subprogram examines pre-allocated locations in the area of PDP-11/34A memory assigned to the DL10 UNIBUS window to obtain values of the parameters that it uses. These are compared with stored values, which reflect the current status of the display, and unless the differences exceed selected threshold values there is no further action. When a change in value exceeds the threshold, the display of the particular instrument is altered to indicate the new value, which is then stored for subsequent comparisons.

The instrument subprograms in turn call graphics primitive subroutines in RGRP11. The primitives for setting the current colour and for drawing a single pixel are coded in-line using macro calls, in the interest of speed. Subroutines are used for drawing vectors*, text, triangular area-fill, and for generating pointers. RGRP11 also contains facilities to aid the creation of the effect of motion in the display.

* Using the same algorithm as described in Section 3.3.2.

3.5.1 Motion

To create the visual effect of motion it is necessary to erase and re-draw the moving element in the picture at high speed. The design of the hardware is such that a part of the display may be erased by setting the controller colour register to match the background (typically black, colour zero) and transmitting to the controller the same sequence of display code (positioning and incremental commands) as was used to draw it. A white triangle on a black field is erased by drawing an equivalent black triangle. However, there is no need to re-compute the display code to draw the black triangle: if the code generated in drawing the white one is stored, then this just needs to be 'played back' to the controller.

RGRP11 facilitates this process. An instrument updating subprogram, drawing say a pointer, can specify a 'private' code storage area, and use versions of the graphics primitives which automatically store the display codes they generate there as well as transmitting them to the hardware. Where a lengthy sequence of code is to be generated and saved, RGRP11 allows the PDP-11/34A hardware stack pointer register to be used as the display code saving pointer instead of a word in memory. The 'auto-decrement indexed' mode of operand addressing can then be used to gain a speed advantage without the need to sacrifice one of the six general purpose registers to code saving. RGRP11 also contains a display code replay subroutine. In a typical instrument updating subprogram this is used to erase the 'old' pointer or numbers when it is determined that the input parameter has changed by more than the threshold. The new display is then generated, and in the process its code saved for the next update.

3.5.2 Area Fill

Most of the instruments on the display contain a triangular pointer, and since any other planar shape can be decomposed into triangles, the triangle was selected as the basic area fill primitive for RGRP11. The vertices of the triangle are specified as x- and y-coordinates of vertex pixels, which are themselves considered to be within the triangle and are therefore always drawn. In the quantized image space of the raster, the location of each pixel's centroid with respect to the triangle's edges, or some approximation thereto, is used to determine whether or not that pixel should be drawn.

Because of the frequency of its use, the triangle fill process should be as fast as possible.

The approach finally adopted is a derivative of the classical span* [10] generation from an ordered edge list. The hardware design determines that methods depending on a capability for the computer to use the frame buffer store as a working memory [11] can not be applied. Exhaustive examination of each pixel to determine if it lies inside the triangle, conceptually the simplest area fill method, takes too long.

The intersections of scan lines with the triangle's edges can be computed from the equations of the edges: this is done (in FORTRAN) in the DECsystem-10 triangle fill subroutine. Implementing such an algorithm on the PDP-11/34A is difficult, because that machine is not equipped with the hardware floating point arithmetic option. Using fixed point arithmetic, for which it is equipped, is a possibility. However the pixel coordinates are specified in nine bits, so that nine bits are required for fractions as well, and the total of eighteen does not fit in the PDP-11/34A word. To avoid multiple precision arithmetic, the size of triangles can be restricted to less than half screen height and width. Doing this allows the differences between coordinates to be represented in eight bits, and the sixteen-bit word provides enough accuracy for the quotients and products.

Triangle fill by this method was implemented in the PDP-11/34A but as it is complex and for efficiency needs more general purpose registers than the six available, an alternative was sought. In the method adopted, a variant of the vector algorithm, described in Section 3.3.2 above, is used to generate tables containing the scan line—edge intersection x-coordinates. These are then joined. This process is not quite straightforward: Figure 8 shows four cases of the edge AB of triangle ABC. The shaded squares denote the pixels generated by the vector algorithm, while those containing a dot are the ones needed to denote the extrema of horizontal spans filling the triangle. In cases a and b, all the pixels identified by the vector algorithm on edge

* A span is defined as a horizontal line segment bounded by two screen end points.

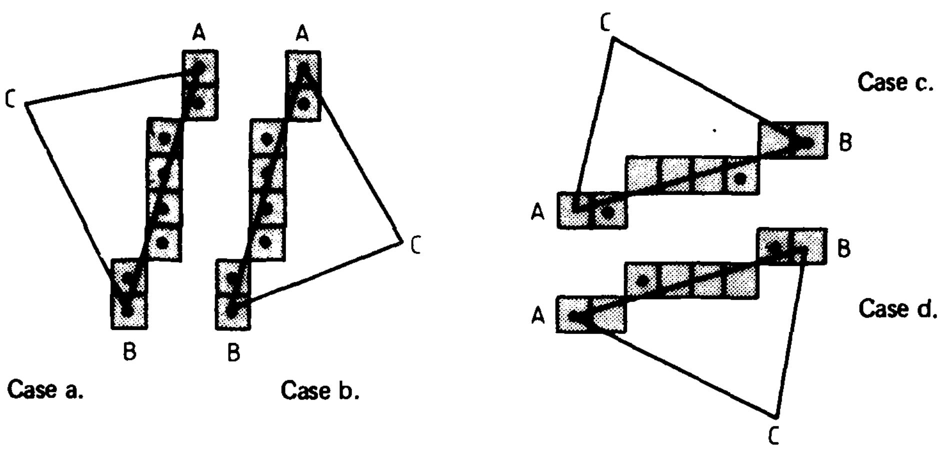


FIG. 8: TRIANGLE EDGE CASES

AB are used, and it can be seen that this is generally true for all edges making an angle of $\pi/4$ or more with the direction of the spans. For constant-y spans, this condition applies when $|\Delta y| \geqslant |\Delta x|$, Δy and Δx being components of the edge resolved in the y- and x-directions respectively.

Cases c and d illustrate the situation when the edge makes an angle less than $\pi/4$ with the spans, or in general when $|\Delta y| < |\Delta x|$ and the spans are constant-y. In these cases not all the pixels generated by the vector algorithm are required, but worse than that, the ones that are required are determined by the geometry of the rest of the triangle. Modification of the vector algorithm to generate the appropriate data is straightforward: the geometry of the triangle can be established from the gradients of (two of) its sides once its vertices have been ordered—ordering by y-coordinate is performed as first step in the triangle fill process. Triangles with horizontal edges are identified as special cases during the ordering.

The resulting triangle fill subprogram in the PDP-11/34A is faster than the computed intersections version, and is more predictable in its behaviour since the standard vector method is used to construct the edges. There are never missed pixels when triangles are juxtaposed in the assembly of other shapes, as can be caused by rounding error with other methods.

3.5.2 Trigonometric Functions

RGRP11 is provided with a look-up table for sine and cosine function values of angles specified in degrees to a resolution of one degree. The function values are represented in single word twos-complement fixed point notation, with the binary point lying to the right of the most significant bit. Thus $-1\cdot0$ is exactly represented (in octal) as 100 000, $0\cdot0$ is exactly 000 000, and $+1\cdot0$ is approximated by the octal value 077 777. A single bit position left-shift operation on the 32-bit product of one of these quantities with a 16-bit integer re-scales the product so that the high 16 bits are the integer and the low 16 an unsigned fraction.

The trigonometric functions are used mostly for converting from polar to rectangular coordinates, to determine say the x- and y-coordinates of the tip of a pointer given its pivot, length and position as an angle.

3.5.4 Text

The same text fonts as in RGRPAC are incorporated in RGRP11, as strings of display code, and there is a subroutine whose function is to draw single characters from these. An added character in each font serves as a rubout for that font (it draws all pixels in the character space) and is used to avoid saving code, see Section 3.5.1 above. Appendix A contains more information on the fonts.

4. IMPLEMENTATIONS

Two of the instruments displayed on the panel, the attitude indicator and the horizontal situation indicator, see Sections 2.2.2 and 2.2.3 above, merit discussion in detail. From these discussions an appreciation of the processes necessary in the implementation of other instruments can be gained.

4.1 Attitude Indicator

The attitude indicator is shown in Figure 9a, with dimensions marked in millimetres. The starting point for the instrument is a background consisting of the outer and inner bezel circles. the bank angle scale marks, and the blue sky semicircle. These are composed off-line using a DECsystem-10 program.

The instrument displays two parameters: pitch and bank angle. These are passed from the DECsystem-10 simulation program as integral numbers of tenths of a degree.

An incremental approach to updating the instrument is required, as it contains so many pixels that their storage (see Section 3.5.1), to say nothing of their re-computation, is a considerable problem.

4.1.1 Horizon and Sky

Figure 9b defines the angles used in this discussion. Taking the case, illustrated, of zero bank angle, the horizon is denoted PQ, with the zenith direction indicated by Z in the sky sector PZQ. Q is on the right when Z is up, and the angles subtended at the centre O by arcs ZQ and ZQP, denoted q and p respectively, are related by the expression (in degrees)

$$p = 360 - q.$$

In turn q is determined by the distance d of chord PQ from O by the formula

$$q = 180 - \cos^{-1}(d/r)$$

where r is the radius of the circle. The distance d is geared directly to the aircraft pitch angle θ, and a ratio of 1 mm per degree was selected—this guarantees that at least five pitch scale bars appear at all times. Working in the upper half of the circle, a table of q values corresponding to the range of d, at steps of 0·5 mm (corresponding to 0·5° pitch increments) was computed. These can be stored in successive PDP-11 8-bit bytes. Values for q in the lower half of the circle (when d has the opposite sign) are supplementary to those in the upper half.

Having determined p and q at zero bank angle, the actual bank angle, ϕ, can be incorporated by subtracting it from both p and q. This is shown in Figure 9c, where the bank angle θ is positive, the aircraft is rolled right wing down, and the horizon has rotated anti-clockwise on the instrument.

Figure 9d shows the general instrument updating problem: the horizon PQ is to be moved to the new position $P'Q'$. Provided the arcs QQ' and PP' are short enough, they can be approximated by straight lines. Considering the (isosceles) triangle OQQ', the distance δ between arc QQ' and chord QQ' is given by

$$\delta = r(1 - \cos \delta q/2).$$

Taking $r = 38$ mm, and $\delta \leqslant 0{\cdot}25$ mm (less than half the smaller, vertical, dimension of a pixel) this gives $|\delta q| \leqslant 13{\cdot}2°$. A value of 10° was therefore selected as the maximum increment in either p or q for a single re-draw. When larger steps occur, intermediate positions for Q' and

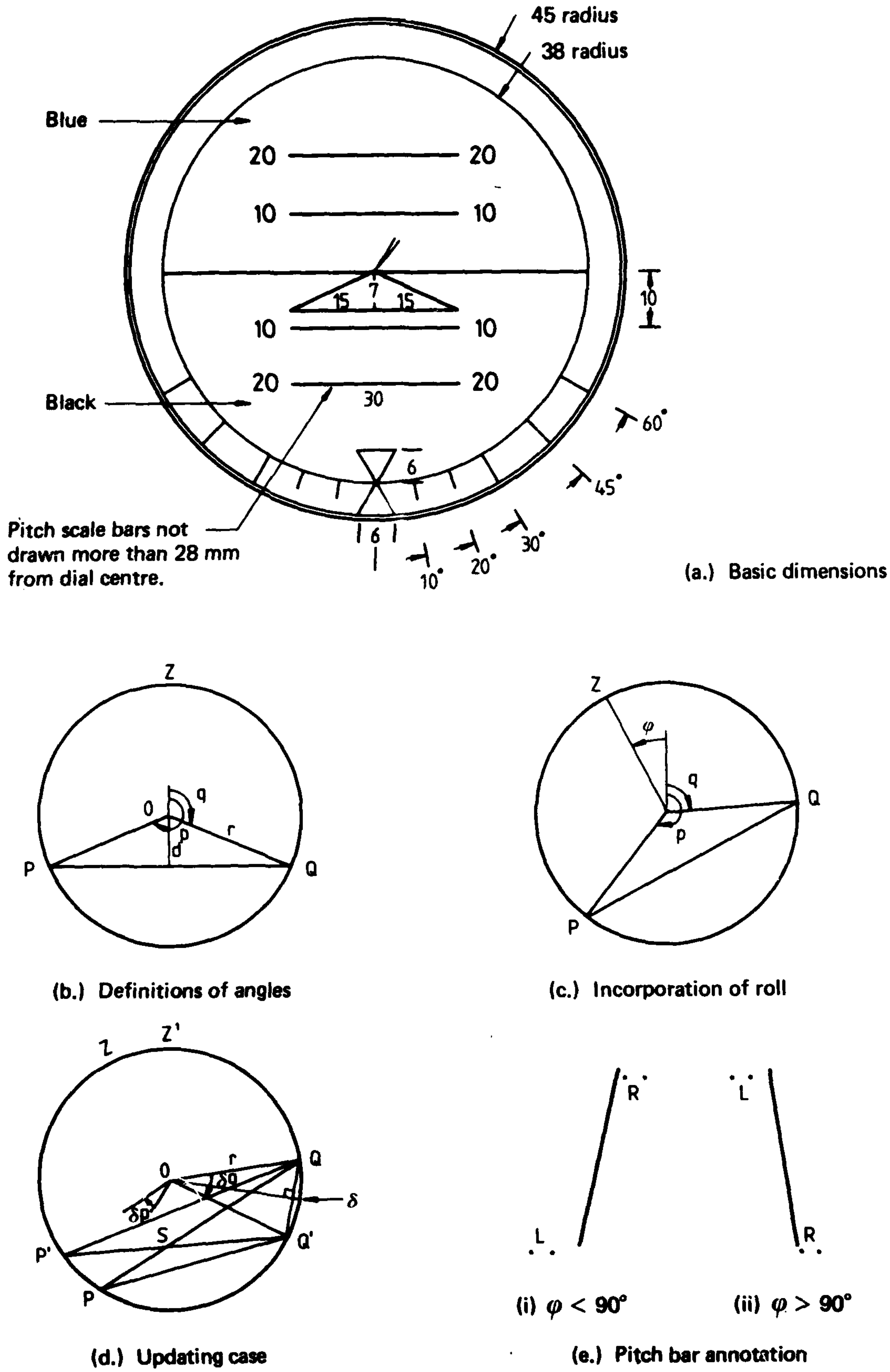


FIG. 9: ALTITUDE INDICATOR CONSTRUCTION

P' are used. In Figure 9d again, the updating can be achieved (assuming for the moment that δq and δp are both smaller in magnitude than 10°) by erasing triangle $PP'S$ and drawing $QQ'S$ in blue. However this requires that the coordinates of the intersection, S, of chords PP' and QQ' be evaluated; and that the (approximations to) line sections $P'S$, SQ' and PS, SQ should not differ noticeably from the (approximations to) lines $P'Q'$ and PQ respectively. This latter condition is not met by the vector algorithm based triangle area fill routine, and the problem can be compounded by rounding error in the evaluation of S. An alternative approach is to delete triangle $PP'Q$ and to draw $QQ'P'$. This is wasteful, to the extent that triangle $P'SQ$ is common to both of these, and is therefore erased and then re-drawn, but there is a saving in the elimination of computation of the coordinates of point S, and of course all the chords are drawn as single best-fit vectors.

The program identifies nine cases, depending on whether p' and q' are greater than, equal to, or less than p and q (respectively), and for each case the appropriate erasing and/or drawing is invoked.

4.1.2 Pitch Scale

The pitch scale bars are located relative to the pitch angle of the aircraft which defines the pitch scale position at the centre of the instrument, so that when the pitch angle is θ the bar at angle b is at a distance $k.(b - \theta)$ in the (un-rolled) y-direction. The proportionality constant k is chosen to be 1 mm per degree. Integer division is used to derive the bar angles in ten-degree steps: thus at pitch angle θ, $(-90° < \theta < 90°)$ the integer quotient $c = \theta \text{ div } 10$, and the truncated pitch angle $d = 10\,c$ are evaluated. Clearly $|d - \theta| < 10°$, and since the pitch scale bars are not drawn further than 28 mm from the instrument centre, i.e. not more than 28° from θ, it suffices to try bar angles

$$b = d \pm 30°,\ d \pm 20°,\ d \pm 10°,\ d.$$

When $|b - \theta| > 28°$ or $b = 0$ (the horizon) the bar is not drawn. The bars are 30 mm long (in the un-rolled x-direction) so that the screen coordinates (x, y) of the right-hand end of the bar at angle b are given by

$$x = 15 \cos \phi - (b - \theta) \sin \phi + x_I$$

$$y = 15 \sin \phi + (b - \theta) \cos \phi + y_I$$

when the bank angle ϕ is included and (x_I, y_I) are the instrument centre coordinates. The coordinates of the left-hand end of the bar are got by changing the signs of the first product in each expression (un-rolled $x = 15$ becomes $x = -15$). The bars are drawn as vectors.

Figure 9e shows how pitch bar annotation is performed, treating values of $\phi > 0$. The annotations are drawn as two numeric characters, relative to the bar end those attached at L in the diagram (indicated by dots) have their origins at $(-10, -1{\cdot}8)$ and $(-5, -1{\cdot}8)$, those at R are at $(1, -1{\cdot}8)$ and $(6, -1{\cdot}8)$. As ϕ passes through 90°, the annotation locations change ends. An identical approach is used when $\phi < 0$ passes through $-90°$.

The pitch bar annotations are limited to 90° at zenith and nadir, running backwards through 80°, 70°, . . . rather than on to 100°, 110°, . . . Since the simulator application under discussion does not permit aerobatic flight, this feature is of small consequence and would, hopefully, be unobserved by most pilots.

4.1.3 Bank Pointer

The bank angle pointer is drawn as a (white) triangle indicating the direction of the nadir. The coordinates of its vertices are computed using analogous rotational transformation formulae to those quoted for the ends of the pitch bars in Section 4.1.2 above, taking maximum advantage of its symmetry to reduce the computations required.

4.1.4 Aircraft Datum

The aircraft datum triangle at the centre of the instrument (see Figs. 2, 9a.) provides the pitch reference for the pilot: its top vertex indicating aircraft pitch angle on the pitch scale

behind it. It overlays the sky, ground and pitch scales—it could be considered to be 'painted on the glass' of a real instrument. To avoid re-computing it each time it is partially erased by background changes, when RGRP11 is initialized code for the triangle is generated once-only and its code saved, and this is re-played (see Section 3.5.1) at the conclusion of each updating cycle.

4.1.5 Order of Operations

The various operations in the updating of the display must be performed in a strict order to preserve its integrity: correct depiction of motion requires that elements must be erased before being drawn in new positions. As a fairly complicated instrument, the order of operations in updating the attitude indicator shows the general approach, and is illustrated in flowchart form in Figure 10.

4.2 Horizontal Situation Indicator

Figure 11a illustrates the general arrangement and dimensions, in millimetres, of the horizontal situation indicator. Coordinates, relative to the instrument centre, of various points are shown. The inner and outer bezel circles, range and track counter boxes and labels, lubber line vee, and tiny aircraft at the centre constitute the background, composed off-line.

The instrument displays four parameters, of which two, range and demanded track, are displayed as numeric readouts in the boxes at the top. The latter also orientates the to-from arrow and its deviation scale relative to the compass card, which is itself rotated to indicate the aircraft's current heading at the lubber line vee at the top of the instrument. Cross-track deviation is displayed by lateral displacement of the to-from arrow.

4.2.1 Compass Card

The compass card is annotated with letters at the cardinal points, two-digit numbers (giving tens of degrees) at the intervening thirty degree intervals, and tic marks at ten-degree intervals between them. Figure 3 shows the annotation in the standard arrangement—the letters and numerals are oriented radially, upright near the top and upside down at the bottom. Figure 11a illustrates the implementation of the compass card: the annotations are all upright, and remain so as the card rotates. This presentation, which appears superior to the standard instrument, is used because the text character fonts are tailored for an upright orientation and are stored as patterns of pixels which are themselves not square.

The method used for determining the position at which the annotations should be drawn is shown in Figure 11b, where the larger circle centred at point C depicts the compass card boundary on which lies the point P, at an angle θ measured clockwise from the vertical, to be annoted. The characters are located by their bottom left-hand corners and point P', used for annotating P, is found by measuring the same angle θ around the circumference of a smaller circle whose centre C' is appropriately offset from C. The diagram shows the dimensions used for determining the positions of the two-digit annotations at the thirty-degree points. Slightly different ones are used for the cardinal point letters.

A similar technique is used to locate the annotations required on other instrument dials, although these are all composed off-line as part of the instrument panel background by a DECsystem-10 program.

Savings in computation time are achieved by exploiting the symmetry of the compass card as shown in Figure 11c. After locating the coordinates of point P relative to the centre O by the formulae $x = r \sin \phi$, $y = r \cos \phi$, the coordinate of the other three points are obtained in turn by repeatedly negating the x-coordinate of the current point and then exchanging the coordinates.

4.2.2 Other Features

The other changing parts of the horizontal situation indicator are the range and track numeric readouts and the to-from arrow and deviation scale. These latter are constructed using

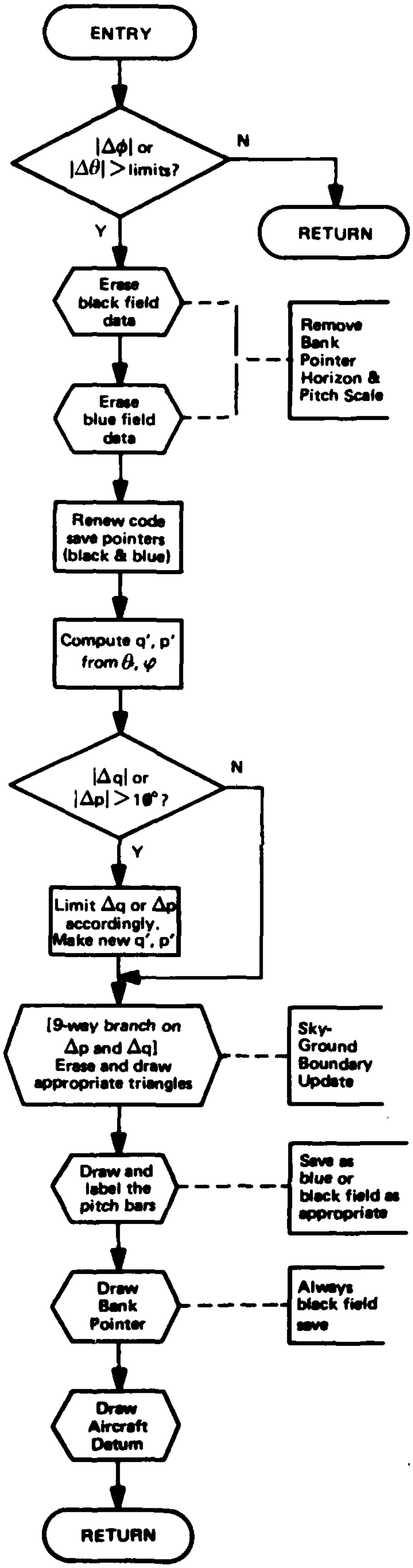
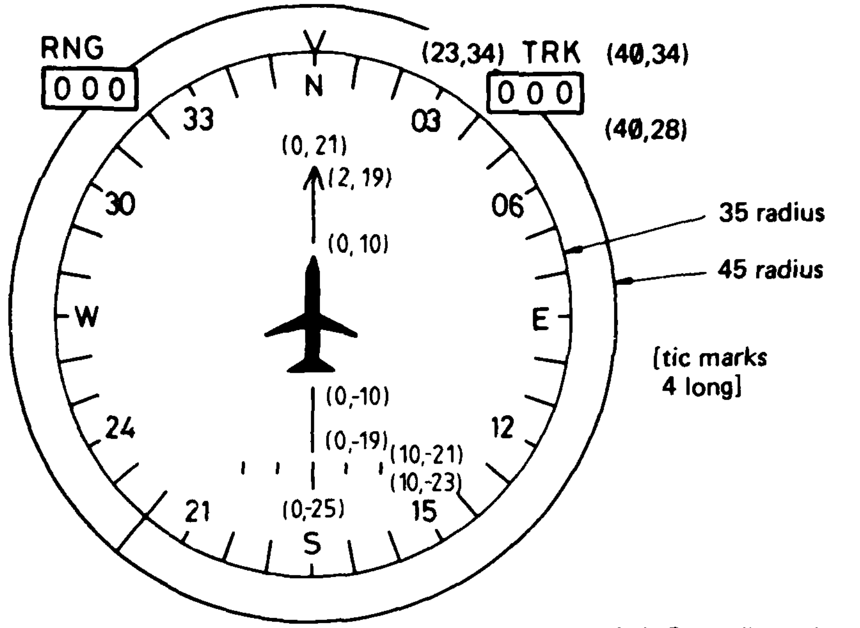
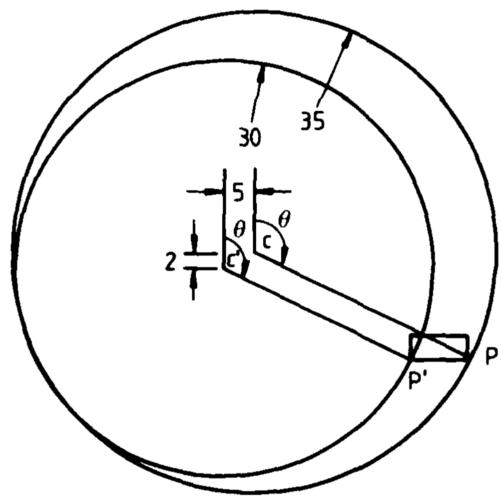


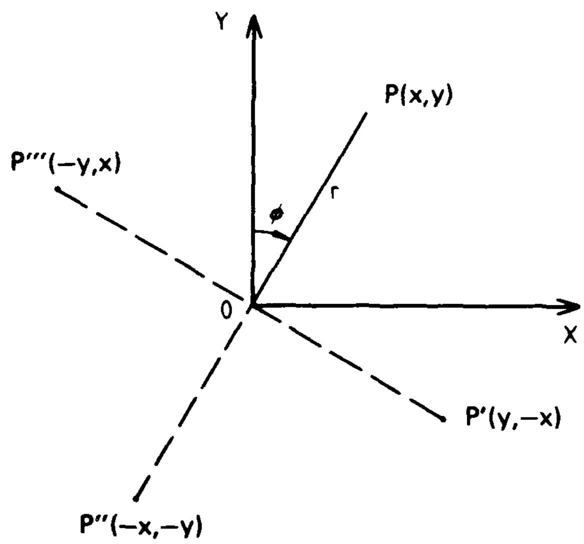
FIG. 10: BANK POINTER ROUTINE OUTLINE FLOWCHART



(a.) Basic dimensions



(b.) Annotation technique



(c.) Symmetry

FIG. 11: HORIZONTAL SITUATION INDICATOR CONSTRUCTION

straightforward [illegible] over a fixed path [illegible] repeated [illegible]

5. RESULTS

[illegible]

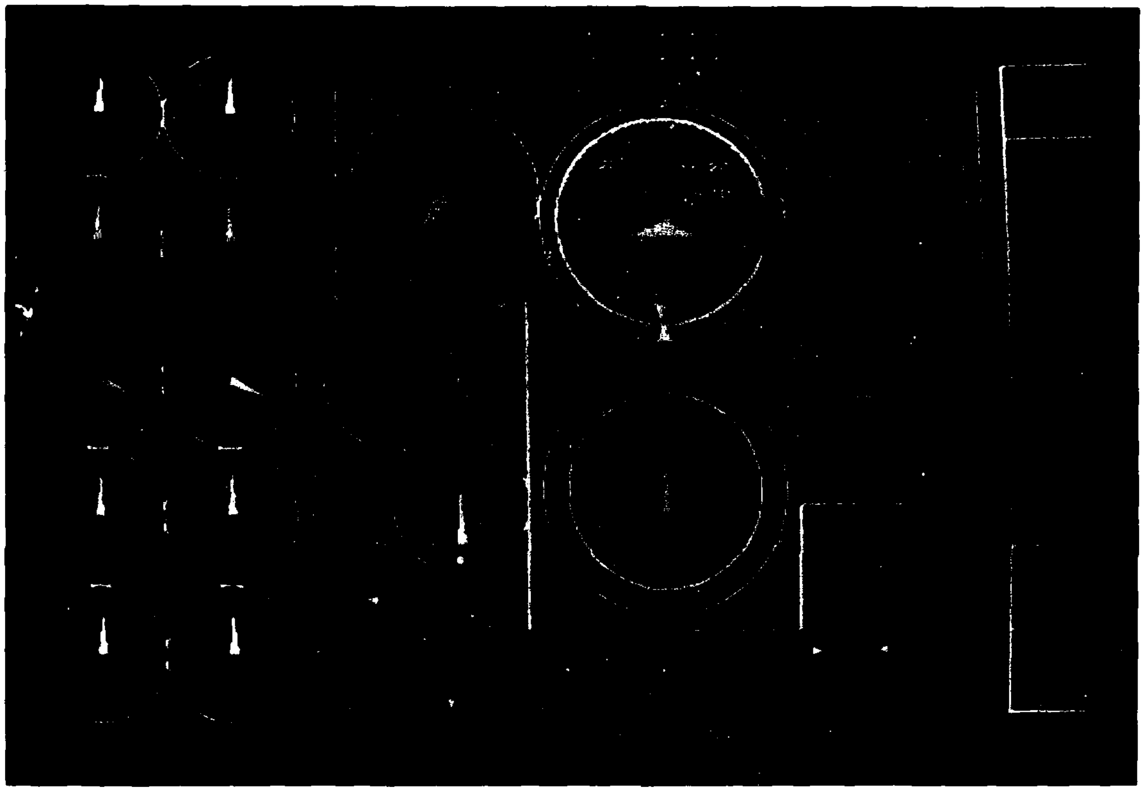
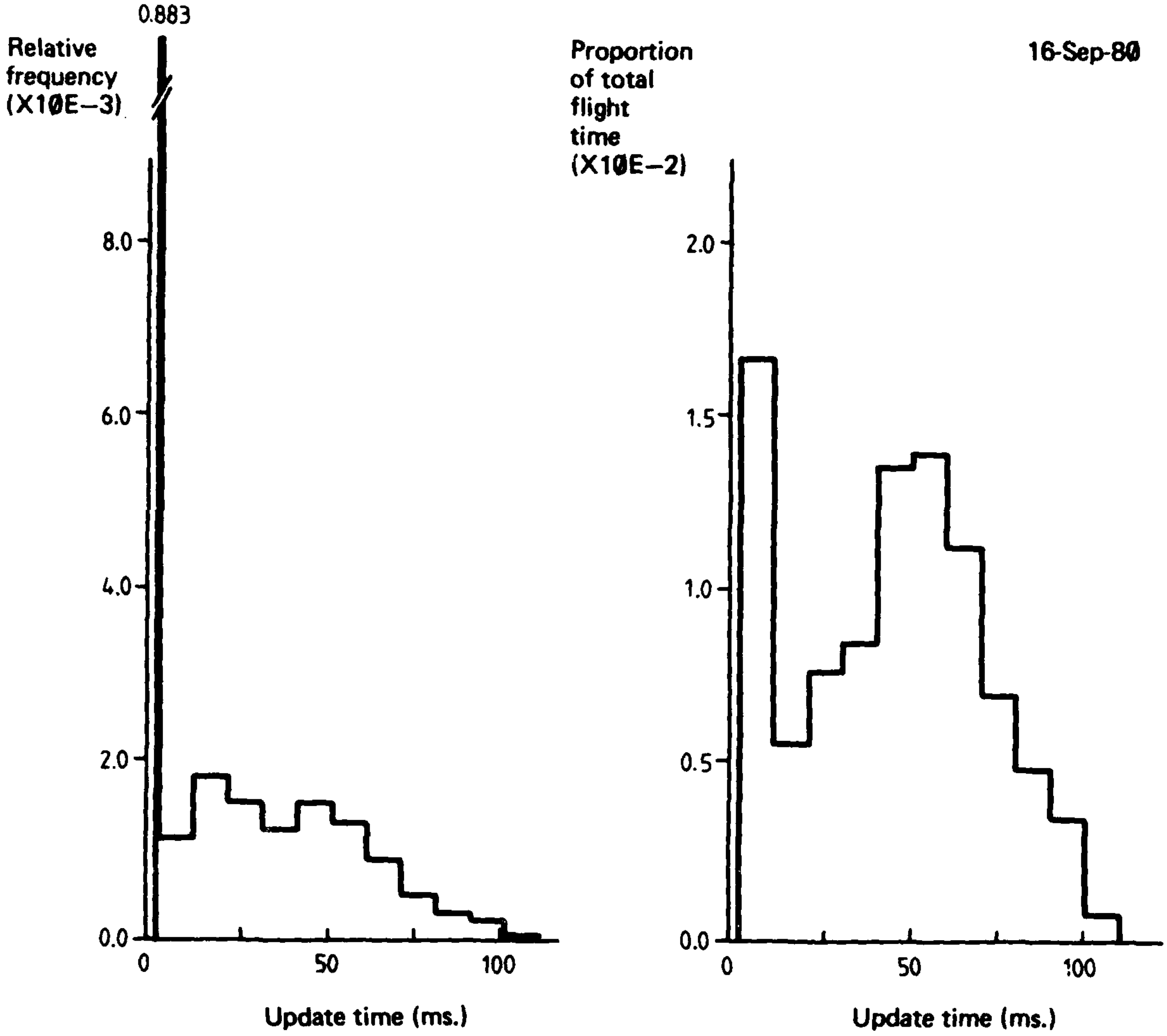


PLATE 1: INSTRUMENT PANEL DISPLAY

Subjective assessments of the representations of the instrument have all been favourable, with civilian and service pilots of a wide variety of backgrounds having successfully flown the simulator using it. At the time of writing, objective measurements of piloted flight performance using the instrument panel display were awaiting completion of verification and documentation of the simulator flight dynamics model. There is [illegible] cause to assume that these tests will be other than satisfactory.

The general approach having been established [illegible], the greatest single uncertainty in the implementation was the ability of the system [illegible] the PDP 11/34A processor [illegible] raster graphics controller combination, to update the display at a high enough rate to allow the aircraft to be controlled by a human pilot. It was quickly verified that the update rate achieved was in fact adequate. The lumped histograms of update time frequencies and proportional occurrence shown in Figure 12, obtained using the PDP-11/34A programmable real time clock to measure the complete execution time of the instrument panel updating loop (Section 3.5 above) during a moderately active flight, confirm that this is so. The relative frequency plots show a large peak near zero; these are traverses of the updating loop made when the display parameters



Maximum update time	= 12Ø ms.
•Minimum update time	= 1 ms.
Total number of updates	= 74179
Total flight time	= 7.16 mins.

FIG. 12: INSTRUMENT PANEL UPDATE TIMINGS

(which are updated twenty-five times per second) show insufficient variation to require any update action. Crudely, the frequency plot is flat past this peak and out to 50 milliseconds and slopes linearly from there to zero at 100 milliseconds, implying that two thirds of the updates where something varies are completed in 50 milliseconds or less, one third take more, with only a tiny proportion exceeding 100 milliseconds. The longer update times occur when the manoeuvres are more violent, and therefore correspond to occasions when accuracy of reading the instruments is not so great. The 120 millisecond maximum update time is an upper limit on the execution time of the complete panel updating loop.

The principal limitation of the raster graphic approach is the quantization of the image, which manifests itself in the steps and stairs aberrations on sloping edges and in the jerkiness

of small motions. This latter effect may give rise to a residual error term when the flight is controlled by a human using the display. Even so, the instruments are quite sensitive, and can probably be read as accurately as 'real' ones; and the existence and magnitude of this residual will need to be established by measurement.

6. DISCUSSION

The potential of the colour raster computer graphic approach in cockpit instrumentation is well demonstrated in this implementation. For conventional instruments its programmable flexibility is the main advantage, but it is in the presentation of entirely novel cockpit information, with complete freedom of formats, that it can be expected to be most useful as part of research simulation activities. Applications for immediate work include investigation of predictive augmentation of pilot's information and microwave landing system-based cockpit displays for unconventional landing trajectory control (curved and varying speed). Both of these will require design and implementation of new formats for the display of the relevant flight parameters and directors. This work has established that there is reason to be confident of the success of the implementation.

The experience gained has also provided much input to the design of a new raster graphics hardware subsystem, particularly in regard to establishing the most appropriate command/data formats. The new hardware will, when completed, be significantly more capable than the existing hardware, while being easier to program for this and similar applications.

REFERENCES

[1] Thelander, H. A.: 'Application of Computer Colour Raster Displays in the Cockpit in Research Flight Simulation', ARL Systems Note 71, March 1980.

[2] —: 'Evolution of the Instrument Panel', Parts 1 and 2, Smiths Industries 'Aviation Review' Vols 32, 34; 1976.

[3] —: 'Layout of Flight Instruments (Aircraft)', Australian Defence Standard DEF (AUST)-398A, October 1969.

[4] —: 'Location and Arrangement of Engine Instruments for Fixed and Rotary Wing Aircraft.' Australian Defence Standard DEF (AUST)-302A, March 1971.

[5] —: 'Standard Display Nomenclature for Cockpit Instruments', Australian Defence Standard DEF (AUST)-468, March 1971.

[6] —: 'Attitude Indicators (Aircraft)', Australian Defence Standard DEF (AUST)-417, August 1969.

[7] —: 'Horizontal Situation Indicator (Aircraft)', Australian Defence Standard DEF (AUST)-414, October 1969.

[8] —: 'Vertical Speed Indicators (Aircraft)', Australian Defence Standard DEF (AUST)-332, May 1968.

[9] Thelander, H. A.: 'Software and System Integration Aspects in Hybrid Computation System Mark 3', ARL Systems Report (in preparation).

[10] Newman, W. M., and Sproull, R. F.: 'Principles of Interactive Computer Graphics', 2nd Ed., McGraw-Hill, New York, 1979.

[11] Ackland, B. D., and Weste, N. N.: 'The Edge Flag Algorithm—A Fill Method for Raster Scan Displays', IEEE Trans. on Computers, Vol. C-30 No. 1; January 1981.

[12] Thelander, H. A.: 'Hybrid Computing System Mark 3 Software Reference Manual', ARL Systems Note 76, January 1981.

APPENDIX A

Alphanumeric Text

A.1 General

Several examples of alphanumeric text fonts designed for use with the plotters and various display devices of the DECsystem-10 are available at ARL. However they are all based on square (equal x- and y-dimension) pixels or grids, and the lateral elongation that they suffer when displayed using the raster graphics hardware described herein makes them unsatisfactory for practical application, although this approach was used in part of the exploratory work described in Reference [1]. Additionally, the hardware design feature providing different colour registers for odd and even pixels (see Section 3.1.4 in the main text above) to allow composite colours to be generated does not fit well with the drawing of characters by means of lines of single pixels. Diagonal lines of single pixels come out as one or other of the constituent colours, whilst horizontal and vertical lines are correctly colour-merged, and the overall effect can be quite odd. This problem is easily avoided by broadening all the lines in characters to be drawn in composite colours out to two pixels width.

The application requires more than one size of characters to be available. To meet this requirement, a DECsystem-10 program was written which allowed full alphanumeric (96-character ASCII) or numeric-only character set generation to be done using the type 338 interactive graphic display. Some discussion of this program, how its output is stored and used, and the characteristics of the fonts developed, follow.

A.2 Font Generation

The program allows the user to specify the size parameters (cell height, width and spacing) of a font when a new font is to be generated, or alternatively to call up an existing or part-completed font for modification or extension. Generating a font can take ten or more hours (elapsed time), so the program also can be stopped, and the font data generated up to that point stored, at the user's convenience. The user interacts with the program using the light pen and terminal. The character code is entered in response to a prompt from the program, whereupon a grid depicting the character space is displayed. The light pen is then used to trace the grid intersections corresponding to pixels in the character, and to indicate those which are to be intensified and those not. Incremental mode (see Section 3.1.4 above) instructions are used throughout. The drawing of the character commences at its bottom left hand corner, and the program requires the user to finish it in the appropriate position for the bottom left hand corner of the following character in a line of text.

A.3 Font Storage

The output from the font generation program consists of the strings of raster graphics controller incremental move instructions required to draw the characters, together with tables defining the location and length of each character's string. It is suitable, after minor editing, for direct incorporation as tables of constants in the program source code of both the DECsystem-10 package RGRPAC and the PDP-11/34A program RGRP11.

A.4 Font Details

Two complete alphanumeric and two numeric only fonts were used in the instrument panel application: *their characteristics are tabulated below:*

Alphanumeric Fonts

Number	Height (Pixels)	Width (Pixels)	Gap (Pixels)	Spacing (mm)	Width (mm)	Height (mm)
0	11	6	3	7·2	4·8	5·9
1	9	5	3	6·4	4·0	4·8

Numeric Fonts

Number	Height (Pixels)	Width (Pixels)	Gap (Pixels)	Spacing (mm)	Width (mm)	Height (mm)
0	7	4	2	4·8	3·2	3·73
1	11	6	3	7·2	4·8	5·9

Numeric font No. 1 is composed in two-pixel wide lines and is therefore suitable for displaying numerals in composite colours.

DISTRIBUTION

	Copy No.
AUSTRALIA	
Department of Defence	
Central Office	
Chief Defence Scientist	1
Deputy Chief Defence Scientist	2
Superintendent, Science and Technology Programmes	3
Aust. Defence Scientific and Technical Rep. (U.K.)	—
Counsellor, Defence Science (U.S.A.)	—
Defence Central Library	4
Document Exchange Centre, D.I.S.B.	5–21
Joint Intelligence Organisation	22
Aeronautical Research Laboratories	
Chief Superintendent	23
Library	24
Superintendent Systems	25
Divisional File-Systems	26
Authors: H. A. Thelander	27
R. J. Rossa	28
P.O. Flight Systems Group	29
Materials Research Laboratories	
Library	30
Defence Research Centre	
Library	31
Central Office	
Director General—Army Development (NSO)	32–35
Central Studies Establishment	
Information Centre	36
Engineering Development Etablishment	
Library	37
RAN Research Laboratory	
Library	38
Navy Office	
Naval Scientific Adviser	39
RAN Air Maintenance and Flight Trials Unit	40
Directorate of Naval Aircraft Engineering	41
Directorate of Naval Aviation Policy	42
Army Office	
Army Scientific Adviser	43
Royal Military College Library	44
US Army Standardisation Group	45

Air Force Office

Aircraft Research and Development Unit, Scientific Flight Group 46
Air Force Scientific Adviser 47
Technical Division Library 48
Director General Aircraft Engineering 49
Director General Operational Requirements 50
HQ Operational Command 51
HQ, Support Command [SESO] 52
RAAF Academy, Point Cook 53

Department of Industry and Commerce

Government Aircraft Factories:
Manager 54
Library 55

Department of Transport

Library 56
Flying Operations and Airworthiness Division 57

Statutory and State Authorities and Industry

Trans-Australia Airlines, Library 58
SEC of Vic., Herman Research Laboratory, Library 59
Ansett Airlines of Australia, Library 60
B.H.P., Melbourne Research Laboratories 61
Commonwealth Aircraft Corporation, Library 62

Universities and Colleges

Adelaide	Barr Smith Library	63
Latrobe	Library	64
Melbourne	Engineering Library	65
Monash	Hargrave Library	66
Newcastle	Library	67
Sydney	Engineering Library	68
N.S.W.	Physical Sciences Library	69
Queensland	Library	70
Tasmania	Engineering Library	71
Western Australia	Library	72
R.M.I.T.	Library	73

CANADA

International Civil Aviation Organization, Library 74
NRC
Aeronautical and Mechanical Engineering Library 75

FRANCE

ONERA, Library 76

GERMANY

Fachinformationszentrum: Energie, Physic, Mathematik GMBH 77

INDIA

Defence Ministry, Aero Development Establishment, Library 78
Hindustan Aeronautics Ltd., Library 79
National Aeronautical Laboratory, Information Centre 80

ISRAEL

Technion-Israel Institute of Technology, Professor J. Singer 81

NETHERLANDS

National Aerospace Laboratory [NLR], Library 82

NEW ZEALAND		
Defence Scientific Establishment, Library		83
Transport Ministry, Airworthiness Branch, Library		84
Universities		
Canterbury	Library	85
SWEDEN		
Aeronautical Research Institute, Library		86
SWITZERLAND		
Armament Technology and Procurement Group		87
F+W (Swiss Federal Aircraft Factory)		88
UNITED KINGDOM		
Royal Aircraft Establishment:		
Bedford, Library		89
Farnborough, Library		90
Dr A. A. Callaway, Farnborough		91
Commonwealth Air Transport Council Secretariat		92
National Physical Laboratory, Library		93
British Library, Lending Division		94
CAARC Co-ordinator, Structures		95
Aircraft Research Association, Library		96
British Aerospace:		
Kingston-upon-Thames, Library		97
Hatfield-Chester Division, Library		98
British Hovercraft Corporation Ltd, Library		99
Short Brothers Ltd, Technical Library		100
Universities and Colleges		
Bristol	Engineering Library	101
London	Professor G. J. Hancock, Aero Engineering	102
Manchester	U.M.I.S.T. Library	103
Nottingham	Science Library	104
Southampton	Library	105
Strathclyde	Library	106
Cranfield Inst. of Technology	Library	107
Imperial College	Aeronautics Library	108
UNITED STATES OF AMERICA		
N.A.S.A. Scientific and Technical Information Facility		109
The John Crerar Library		110
Allis Chalmers Corporation, Library		111
Kentex Research Library		112
Lockheed-California Company		113
Lockheed Missiles and Space Company		114
Lockheed Georgia		115
McDonnell Aircraft Company, Library		116
Universities and Colleges		
Johns Hopkins	Library, Applied Physics Laboratory	117
M.I.T.	Library	118
Spares		119–128

LMED

-8